MCP and A2A

I. MCP

1. Defenition

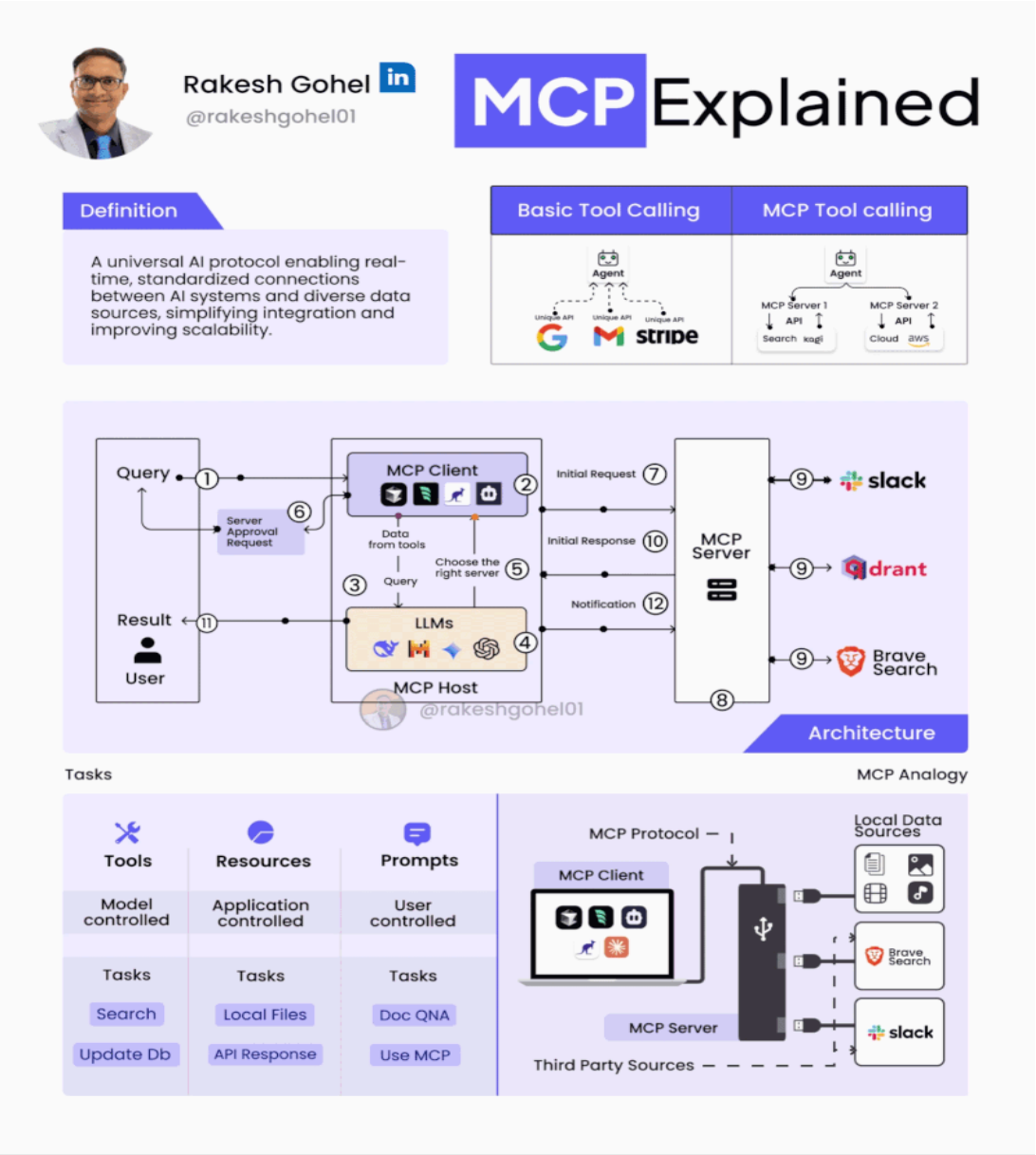
MCP là viết tắt của Model Context Protocol, có thể hiểu là:

- một giao thức kết nối AI với dữ liệu theo chuẩn thống nhất.
- Nhằm: tích hợp đa dạng nguồn dữ liệu, tăng khả năng mở rộng, đảm bảo tính chuẩn hóa, dễ tái sử dụng.

Nhìn chung đây là một giao thức giúp cho việc LLM tool-calling trở nên có tổ chức và dễ mở rộng hơn, tránh hard-code.

2. Các thành phần

Một MCP có một vài thành phần chính và quan trọng nhất là MCP Host, MCP Client, MCP Protocol, LLM và MCP Server. Dưới đây là Analogy của một hệ thống MCP: (theo Rakesh Gohel):



MCP Host: là thành phần cấp cao nhất (kiểu ứng dụng giao tiếp giữa người dùng và hệ thống), có thể chứa 1 hoặc nhiều MCP Client, nó kiểu như chat app, IDE…

- MCP Host kết nối với 1 hay nhiều MCP Server, thông qua MCP Protocol (đây kiểu lớp trung gian ở giữa).
- Là I/O của hệ thống.

MCP Client nằm trong MCP Host, nhiệm vụ của nó là nhận query từ MCP Host (nơi người dùng đã hỏi gì đó). Nó cần có:

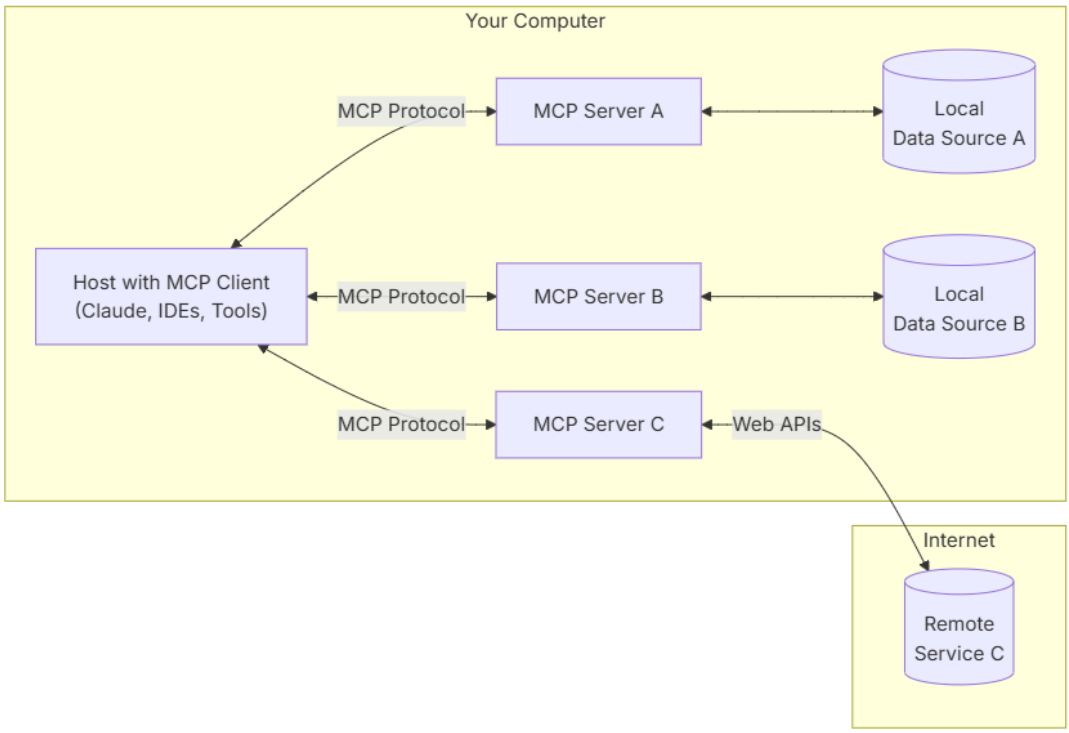
- Quản lý JSON schema liên quan đến các tools và servers có thể sử dụng. (các cái tool description nằm ở trên MCP Server, Client cần lấy được nó để biết server nào có tool nào).
- Nó gửi truy vấn cùng với các schema đó đến LLM, từ đây LLM sẽ quyết định tools nào cần sử dụng, server nào cần…, sau đó gửi lại cho MCP Client (số 5).
- Có được điều này vì ta coi như mỗi server phục vụ 1 domain, thì từ truy vấn, LLM cần quyết định xem phải dùng tool nào ở server nào, và gửi lại payload JSON cho Client để nó còn request.

MCP Server: MCP Server là nơi xử lý truy vấn thực tế, đóng vai trò như hub kết nối đến các nguồn dữ liệu hoặc công cụ bên ngoài, ví dụ: Cơ sở dữ liệu (SQL/NoSQL), Dịch vụ web/API. Search engine (Brave Search)

- Có mô tả về các tools, tham số tools đó cần, để client có thể lấy được và gửi cho LLM xây quyết định.
- Khi Client gửi request đến Server:
    - Kiểm tra payload đầu vào có hợp lệ theo schema không.
    - Gọi công cụ tương ứng.
    - Trả kết quả cho Client (hoặc mở thêm các bước trao đổi tiếp theo nếu cần, nó có thể gọi nối tiếp đến các MCP Server khác).

MCP Protocol: như một kiểu chuẩn hóa cách client giao tiếp với server. (theo mình hiểu nó như một kiểu format chuẩn thôi, ví dụ cấu trúc json etc…).

Có thể hiểu đơn giản hơn qua ví dụ của Anthropic:



3. ***Luồng hoạt động:*** Gần như đã được thể hiện hết trong cheetsheet bên trên rồi, dưới đây là ví dụ mô tả về luồng hoạt động đó:

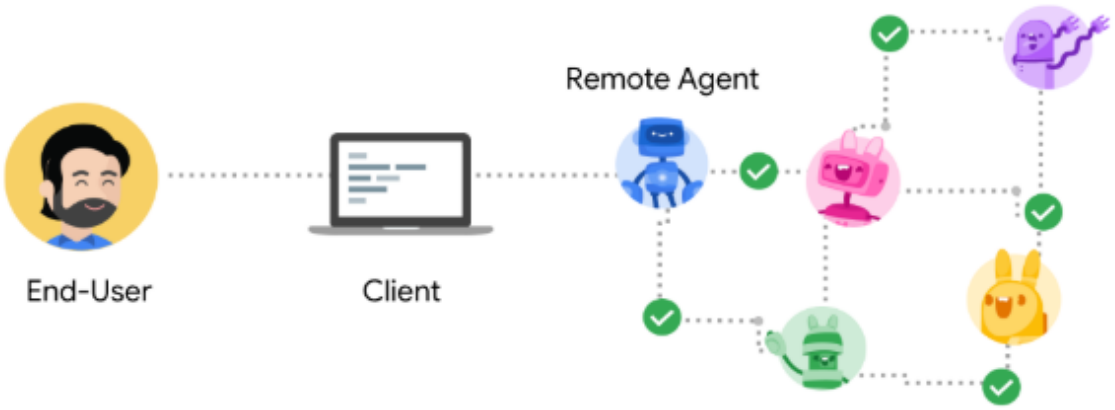
    (1) Người dùng nhập input "truy vấn cho tôi số khách hàng tại hà nội" -> client nhận câu hỏi
    (2) Client gọi các server lấy list tools tương ứng với từng server. Sau đó nó ném query + tools schema đó cho LLM.
    (3) LLM quyết định gọi tool nào, với tham số nào, ở server nào, ném lại payload json cho client.
    (4) Client lấy payload đó request lên server, lấy response, rồi trả ngược lại.

-> tóm lại: MCP giúp phần connect giữa Agent và các tools + data dễ dàng hơn.

II. A2A (Agent-to-Agent)

1. Khái niệm

Là một tiêu chuẩn mở mới được phát triển bởi google nhằm mục tiêu giải quyết một thách thức cơ bản trong lĩnh vực AI: làm thế nào để các tác nhân AI (AI agents), được xây dựng bởi các đội ngũ khác nhau, sử dụng các công nghệ khác nhau và thuộc sở hữu của các tổ chức khác nhau, có thể giao tiếp và cộng tác hiệu quả?



Như hình ví dụ, thường cái case sử dụng của nó là 1 Client Agent sẽ giao tiếp với Remote Agent khác. Và các Agent này có thể đến từ bất cứ framework nào, không quan trọng, chúng chỉ cần tuân thủ theo A2A là có thể giao tiếp được. Điều này rất lợi là vì:

- Nhiều công ty khác nhau thích dùng framework khác nhau, khi dùng sản phẩm của họ k thể bắt họ đổi được.
- Linh hoạt để mở rộng hơn, có thể tích hợp với bất cứ Agent nào khác một cách dễ dàng.

Ý tưởng chính: Mỗi agent sẽ có một endpoint http (để call đến) và một "danh thiếp", mô tả xem nó đảm nhận chức vụ gì và có thể làm được gì. Sau đó khi công việc đến, ta điều phối công việc tương ứng với nhiệm vụ của từng agent được ghi trên "danh thiếp". Sẽ được nói rõ hơn ở các phần sau.

2. <u>Các concepts cần biết thêm:</u>

Agent Card: Như là một danh thiếp cho các Agent, ở đó chứa thông tin Agent đó có thể làm gì, cần gì, bằng một format JSON. Nói cách khác nó kiểu meta data cho mỗi agent.

- Nói chung nó chứa hết mọi thứ như mô tả, I/O, yêu cầu authen, skills của mỗi agent.
- Nhưng cần một đường dẫn chuẩn cho danh thiếp của mỗi agent, kiểu `/well-known/agent.json`.

A2A Client: Cầu nối giữa User và A2A Server, là một AI Agent hoặc ứng dụng khác:
- Lấy thông tin của các Agent khác từ Agent card.
- Chuyển yêu cầu của người dùng thành chuẩn định dạng A2A, sau đấy điều phối cho Remote Agent tương ứng có thể xử lý được yêu cầu, rồi nhận phản hồi.

A2A Server: Như một hệ thống AI Agent bình thường, được host chạy được, có endpoint https, nhận JSON từ A2A Client, cook và response hoặc hỏi thêm câu hỏi phụ nếu cần.

A2A Task: Là đơn vị công việc chính trong A2A, nhìn chung nó là để Client cho mấy con server biết nhiệm vụ là gì.
- Client sẽ **khởi tạo một task** để đạt một mục tiêu cụ thể (ví dụ: "tạo báo cáo," "đặt vé máy bay," "trả lời câu hỏi"). Nó sẽ tạo ra một ID riêng cho từng task.
- Mỗi task có lifetime rõ ràng, nó được thể hiện qua một vài status như `submitted`, `working`, `input-required`, `completed`, `failed`.

- Mỗi một `task` có thể cần nhiều turn trao đổi giữa Client và Server để hoàn thành.

A2A Message: Là một turn giao tiếp trong quá trình thực hiện 1 task. Tức là mỗi task có thể có nhiều message giữa client và server.
- Cấu trúc: `role: user/agent` là client gửi hoặc server trả lời,
- `Parts`: danh sách các phần nội dung. Mỗi parts sẽ bao gồm type và nội dung bên trong nó, sẽ được ví dụ ở bên dưới. Đây cũng là đơn vị nhỏ nhất, một Message cũng có thể có nhiều Parts.

A2A Artifact: Là kết quả đầu ra Server cook xong sau đó ném cho A2A Client, bên trong nó cũng có thể có nhiều `Parts.`

Các ví dụ có thể tìm thấy trong docs gốc của Google, được đề cập dưới kia!

III. Sources

Dưới đây là một số luồng chính mình đã tham khảo, tất nhiên còn 1 vài nguồn phụ từ thảo luận hay một số blog mình không nhớ hết tên :D

1. MCP:

Rakesh Gohel Blog:
https://www.linkedin.com/posts/rakeshgohel01_if-i-had-to-start-learning-mcp-from-scratch-activity-7325499211976736772-x6yL?utm_source=social_share_send&utm_medium=member_desktop_web&rcm=ACoAAEr3bOQBRLIDlhcUVvccliUG3Pbnpx_C4Lk

Docs của Anthropic: https://www.anthropic.com/news/model-context-protocol
Youtube Explain của IBM: https://www.youtube.com/watch?v=eur8dUO9mvE

2. A2A:

Google: https://google.github.io/A2A/topics/key-concepts/#2-core-concepts-summary
Link youtube hữu ích: https://www.youtube.com/watch?v=mDcULe4GMyA